Evaluating Neural Word Embeddings Created from Online Course Reviews for Sentiment Analysis

Danilo Dessì
University of Cagliari
Cagliari, Italy
danilo_dessi@unica.it

Mauro Dragoni
Fondazione Bruno Kessler
Trento, Italy
dragoni@fbk.eu

Gianni Fenu
University of Cagliari
Cagliari, Italy
fenu@unica.it

Mirko Marras
University of Cagliari
Cagliari, Italy
mirko.marras@unica.it

Diego Reforgiato Recupero
University of Cagliari
Cagliari, Italy
diego.reforgiato@unica.it

ABSTRACT

Social media are providing the humus for the sharing of knowledge and experiences and the growth of community activities (e.g., debating about different topics). The analysis of the user-generated content in this area usually relies on Sentiment Analysis. Word embeddings and Deep Learning have attracted extensive attention in various sentiment detection tasks. In parallel, the literature exposed the drawbacks of traditional approaches when content belonging to specific contexts is processed with general techniques. Thus, ad-hoc solutions are needed to improve the effectiveness of such systems. In this paper, we focus on user-generated content coming from the e-learning context to demonstrate how distributional semantic approaches trained on smaller context-specific textual resources are more effective with respect to approaches trained on bigger general-purpose ones. To this end, we build context-trained embeddings from online course reviews using state-of-the-art generators. Then, those embeddings are integrated in a deep neural network we designed to solve a polarity detection task on reviews in the e-learning context, modeled as a regression. By applying our approach on embeddings trained using background corpora from different contexts, we show that the performance is better when the background context is aligned with the regression context.

KEYWORDS

Big Data, Deep Learning, Online Education, Sentiment Analysis, Word Embedding.

ACM Reference Format:

Danilo Dessì, Mauro Dragoni, Gianni Fenu, Mirko Marras, and Diego Reforgiato Recupero. 2019. Evaluating Neural Word Embeddings Created from Online Course Reviews for Sentiment Analysis. In *The 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19), April 8–12, 2019, Limassol, Cyprus.* ACM, New York, NY, USA, Article 4, 5 pages. https://doi.org/10.1145/3297280.3297620

SAC '19, April 8–12, 2019, Limassol, Cyprus

ACM ISBN 978-1-4503-5933-7/19/04.
https://doi.org/10.1145/3297280.3297620

1 INTRODUCTION

On a daily basis, people express their opinions about products, services and facts through online platforms in various contexts such as e-commerce sites, discussion boards, e-learning platforms, and so on. The automatic recognition of sentiments and emotions in such opinions has gained a great attention in academia and industry. The process of detecting subjective insights from a texts using Natural Language Processing (NLP), text mining and computational linguistics is called Sentiment Analysis [6]. This field has been recently improved with the employment of semantic approaches and resources [16]. One of the most known tasks in Sentiment Analysis is the polarity detection: given a collection of text documents, the goal is to correctly infer their discrete or continuous polarity.

Emerging methods leveraged word embeddings generated from a large collection of documents, showing different ways and datasets for their construction [10, 13, 14]. Word embeddings represent the contextual information of a given corpora and capture syntactic and semantic information with respect to the dataset used for building the embeddings. Their adoption showed improvements in the Sentiment Analysis area [8]. Unfortunately, models built upon these generic-trained word embeddings demonstrated to under-perform the ones built upon word embeddings created from textual resources coming from the same context of the Sentiment Analysis task [5]. Hence, it becomes crucial analyzing context-trained embeddings and comparing them with generic-trained embeddings.

Online education is one of the contexts receiving great attention [3]. The related platforms can be envisioned as dedicated social networks where discussions focus on specific topics concerning the course content quality, the teachers' skills, and so on [2]. Several Sentiment Analysis tasks related to the students' opinions have been designed for recommending content or highlighting deficiencies in courses [7]. The social network structure and interaction dynamics interfere in the Sentiment Analysis and the Sentiment Analysis can benefit from the fact that learners interact in a social network. Therefore, analyzing their comments as separated piece of information and discovering relationships among comments become interesting. For instance, the automatic understanding of the polarity behind a review can help learners assess the course quality and decide to attend it. The power of a social network in this context is that the actions performed by a student (e.g., writing reviews) can influence and be influenced by other students.

In this paper, we move the first step towards a smart system for analyzing the social network in e-learning and the social effects generated by course reviews left by learners. To this end, we propose a Sentiment Analysis approach that assesses the polarity of a review. First, we created word embeddings tailored to our context through state-of-the-art generation algorithms and data coming from a real-world online course review dataset [4]. Then, the embeddings were fed in a neural network we designed to predict scores that reflect the satisfaction of learners on courses. We evaluated the feasibility of our context-trained embeddings and compared our approach against state-of-the-art baselines. Our contribution is twofold:

- The validation of our deep learning approach for polarity detection with different word embeddings trained on the e-learning context.
- The evaluation of the feasibility of contextual sentiment models and the comparison of the effectiveness of several regression methods for polarity detection in e-learning.

The results showed that our approach powered by contextual embeddings outperforms several baselines on a real-world dataset.

The paper is organized as follows. Section 2 presents the related work. Section 3 describes our approach. Results and discussions are shown in Section 4. Finally, Section 5 concludes the paper.

2 RELATED WORK

Word embeddings have been greatly and widely employed for Sentiment Analysis. However, traditional methods for their construction did not usually consider word distributions for a specific task. To mitigate this drawback, [11] incorporated prior knowledge at both word and document level to investigate the influence each word has on the sentiment label of both target word and context words. On the same direction, [19, 20] employed text sentiment for generating words embeddings. They combined context and sentiment level evidences, so that the nearest neighbors in the sentiment embedding space are semantically similar. The rationale behind that comes from the fact that words with similar contexts but opposite sentiment polarity, such as good and bad, are mapped to neighboring word vectors. Similarly, [22] integrated sentiment information into semantic word representations and extended Continuous Skip-gram model, showing that the learned sentiment word embeddings captured sentiment and semantic information. Furthermore, [12] used unsupervised and supervised techniques to learn word vectors capturing semantic term-document information and sentiment content. The model outperformed several methods for sentiment classification. In the Twitter context, [21] learned sentiment specific words embeddings through three neural networks incorporating the supervision from sentiment polarity. Moreover, [18] integrated word embeddings for the estimation of levels of negativity in plenary speeches, showing that the word embeddings approach has a potential for Sentiment Analysis in social sciences. Several challenges have been also created to solve the polarity detection task and the winning systems mainly employed word embeddings [17].

3 THE PROPOSED APPROACH

This section shows our approach for polarity detection (Figure 1).

3.1 Review Splitter

The Review Splitter receives a set of reviews R, each including a comment t and a rating class $s \in S = \{s_1, ..., s_{|S|}\}$. It accepts a value N defining how many reviews for each class in S are chosen for train/test, and a value $M < N$ defining how many reviews from the N reviews selected for each class are used for train. From R, the Embedding Set Splitter randomly chooses N samples for each class in C and puts them into R_{23}. The other reviews represent $R1$. The comments in $R1$ are concatenated in a text corpus T fed into the Neural Word Embedding Generator. The Training/Testing Set Splitter randomly gets M samples from R_{23} for each class in S, and puts them in $R2$. The others represent $R3$. Finally, $R2$ and $R3$ are fed into the Training and Testing Pre-Processor, respectively.

3.2 Neural Word Embedding Generator

The Neural Word Embedding Generator takes a text corpus T and returns a set of feature vectors E, each representing the word embedding for a given word in that corpus. The feature values are non-negative real numbers. The component also accepts input parameters to specify the algorithm for generating word embeddings, the word embedding size, and the number of context words the generator looks at. The Embedding Selector can choose the underlying generator among *Word2Vec* [13], *GloVe* [14] or *FastText* [10].

3.3 Review Pre-Processors

The Training Pre-Processor takes a set of reviews $R2 = \{(t_1, s_1), ..., (t_{|R2|}, s_{|R2|})\}$, where each pair (t_i, s_i) identifies a text comment t_i and the rating s_i. Then, it returns a set of pre-processed reviews $R2' = \{(v_1, s_1), ..., (v_{|R2|}, s_{|R2|})\}$, where each pair (v_i, s_i) includes an integer-encoded vector v_i of the text comment t_i and the original rating s_i. To this end, the Training Pre-Processor uses a function $f : W \rightarrow \{0, ..., |W| - 1\}$ to uniquely map each distinct word w in the vocabulary W to an integer value in the range $[0, |W| - 1]$. Then, for each comment t_i, the Training Pre-Processor builds an integer-encoded vector v_i, where v_{i_j} represents the integer value mapped by f for the word t_{i_j}. Considering a sample comment t_i = "it was bad" and a function f which maps "it" to 34, "was" to 27 and "bad" to 103, the integer-encoded vector v_i for t_i is [34, 27, 103]. Finally, $R2'$ is passed to the Deep Neural Network Trainer. The same procedure is repeated by the Testing Pre-Processor. It gets $R3$ as input and passes $R3'$ to the Deep Neural Network Regressor.

3.4 Deep Neural Network Trainer & Regressor

The Deep Neural Network Trainer receives word embeddings E and pre-processed reviews $R2'$. With them, the component trains a deep neural network which is then passed to the Deep Neural Network Regressor (DNNR). The latter takes a pre-trained network and a comment t from $R3'$ and returns the predicted sentiment score s. The architecture of the neural network is inspired by [1]. Differently from them, we adopted only one Bidirectional LSTM (Long Short-Term Memory) layer to improve the efficiency and we setup the last layer to return a continuous value as sentiment score.

The Input Layer accepts integer-encoded vectors built by the Review Pre-Processors. These vectors are then passed to the Embedding Layer which organizes them as a matrix of shape (N, M), where N is the number of integer-encoded comments, while M the

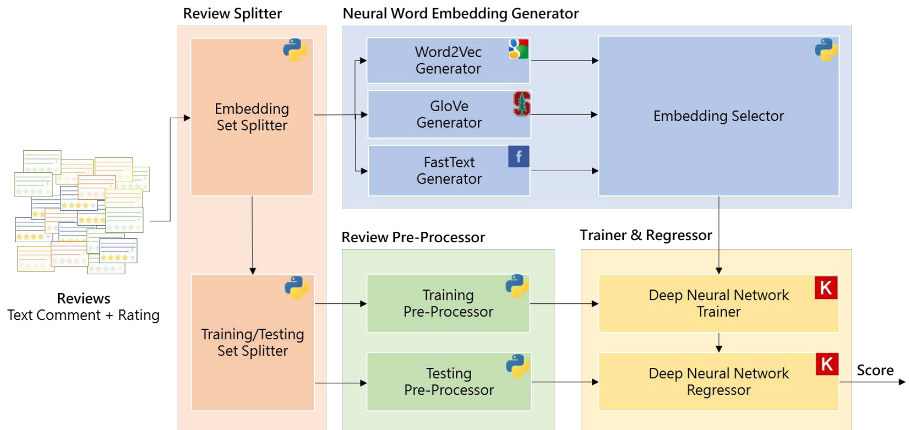

Figure 1: The proposed deep learning approach for performing regression in polarity detection through word embeddings.

maximum length of each integer-encoded vector. Each row is an integer vector representing a given comment. For each one of them, the output of the Embedding layer is a two-dimensional vector, each row representing the word embedding for the corresponding integer-encoded word in the input comment. Before receiving data, the Embedding Layer loads the pre-trained word embeddings computed by the Neural Word Embedding Generator as weights.

The Bidirectional LSTM Layer extends the traditional LSTM [9] by training two LSTMs instead of just one: the first is trained on the input sequence as it is (Forward LSTM) and the second on a reversed copy (Backward LSTM). Combining the two hidden states, the layer can preserve information from both past and future, understanding better the context. Given a sample comment "Very informative and useful course. Easy to understand.", with Bidirectional LSTM, the network gets insights from two sides (e.g., Forward LSTM sees "Very informative and", Backward LSTM sees "Easy to understand"). In this way, it is easier for the network to predict the polarity. Forward and backward outputs are concatenated and returned as output.

The Attention Layer is based on the implementation given by [15]. It enables the network grasp the words of the comment which are most informative at a given stage for predicting the polarity score. To this end, this layer learns a weight for each word of the input comment, expecting keywords to have a heavier weight, and less-informative words to have a lighter weight. The weight of the word reflects its contribution to the polarity of the comment.

Finally, the Dense Layer is a regular densely-connected layer providing a single output unit which represents the polarity score. The network measured the mean squared error of the predicted scores against the target values for 128-sized batches on 20 epochs.

4 EXPERIMENTAL EVALUATION

In this section, we evaluate the performance of our approach developed in Python using Keras[1] on top of a NVIDIA Titan Xp GPU[2].

[1]https://keras.io/

[2]The Titan Xp used for this research was donated by the NVIDIA Corporation.

4.1 Dataset and Metrics

The experiments leveraged COCO [4], a large-scale dataset with 43K courses and 2,5M learners who left 4.5M reviews and related ratings within a scale of 10 discrete values. The Review Splitter considered 1.396.312 English reviews with N=6500 and M=650. For each test, we performed 10-fold stratified cross validation and measured MSE (Mean Squared Error) and MAE (Mean Absolute Error).

4.2 Evaluating Deep Neural Network Regressor

This experiment aims to show how our neural network has advantages over machine learning and multilayer perceptron regressors, when our context-trained word embeddings are used as features.

Baseline Regressors. We compared our approach against Support Vector Machines (SVR), Random Forests (RF), and Multi Layer Perceptron (MLP) regressors in Scikit-Learn[3]. For machine-learning regressors, word embeddings of words in a given text comment were averaged to represent each comment with a single vector.

Results and Discussion. Inspecting Table 1, the baseline algorithm which performed better is MLP, but it was outperformed by our deep learning approach considering every type of neural word embedding. The largest difference between our approach and baselines methods is *1.896* for MSE (*RF + Word2Vec* against *DNNR + Word2Vec*) and *0.436* for MAE (*RF + Word2Vec* against *DNNR + Word2Vec*), whereas the smallest difference is *0.178* for MSE and *0.021* for MAE (*MLP + FastText* against *DNNR + FastText*). For the same methods adopted to create the word embeddings, our neural network approach obtains the best performance. The main advantage might depend on the bidirectional LSTM layers that allows our model to explore data in forward and backward directions, detecting patterns that the proposed baseline approaches ignore.

4.3 Evaluating Contextual Embeddings

This experiment aims to show how the context-trained word embeddings have advantage over generic-trained ones, when fed into our neural network as frozen weights of the Embedding Layer.

[3]http://scikit-learn.org/stable/index.html

Table 1: Comparative analysis of our approach against baseline algorithms with context-trained word embeddings.

Regressor	Embedding Generator	MSE	MAE
RF	Word2Vec	5.248	1.849
	GloVe	5.473	1.897
	FastText	5.170	1.838
SVR	Word2Vec	4.174	1.627
	GloVe	5.380	1.910
	FastText	5.347	1.920
MLP	Word2Vec	4.060	1.585
	GloVe	4.266	1.632
	FastText	3.995	**1.527**
DNNR	Word2Vec	**3.352**	**1.413**
	GloVe	**3.851**	**1.544**
	FastText	**3.817**	1.548

Table 2: Comparative analysis of our context-trained word embeddings and existing generic-trained word-embeddings.

Type	Embedding Generator	MSE	MAE
Contextual	Word2Vec	**3.352**	**1.413**
Generic		4.584	1.729
Contextual	GloVe	3.851	1.544
Generic		**3.785**	**1.543**
Contextual	FastText	**3.817**	**1.548**
Generic		4.713	1.733

Baseline Generic-Trained Embeddings. We performed experiments using 300-sized embeddings trained on COCO, comparing them against the following 300-sized generic-trained word embeddings adopted in literature: *Word2Vec*[4], *GloVe*[5] and *FastText*[6].

Results and Discussion. Table 2 shows how context-trained word embeddings get lower error values in predicting scores. The best results were obtained by context-trained word embeddings generated by *Word2Vec*. The advantages of using context-trained word embeddings is relevant with a maximum of *3.817* and a minimum of *3.352* of MSE against a maximum of *4.713* and a minimum of *3.785* of MSE for generic-trained word embeddings. At the same time, the maximum MAE is *1.548* and the minimum MAE is *1.413* with context-trained word embeddings against a maximum MAE of *1.733* and a minimum of *1.543* with generic-trained ones. These results indicate how context-trained word embeddings are suitable to train models that can predict target scores closer to real ones.

5 CONCLUSIONS AND FUTURE WORKS

In this paper, we moved the first step towards a smart system for social network analysis in e-learning by leveraging the feedback left by learners after attending courses. We created context-trained word embeddings from course reviews and fed them in a neural network to solve a polarity detection task in the same context. The results showed that the generated context-trained word embeddings are suitable for the presented task and our approach powered by them outperforms state-of-the-art baselines on a real-world dataset.

In next steps, we plan to (i) analyze other ways to generate context and sentiment-aware word embeddings, (ii) examine the difference among context/general-trained embeddings, which words their embeddings change significantly and have strong impact in the polarity, and (iii) explore the social effects generated by reviews.

ACKNOWLEDGMENTS

Dessì and Marras acknowledge Sardinia Regional Government for the financial support of their PhD scholarship (P.O.R. Sardegna F.S.E. Oper. Progr. of the Autonomous Region of Sardinia, European Social Fund 2014-2020 - Axis III "Education and Training", S.G. 10.5).

REFERENCES

[1] M. Atzeni and D. Reforgiato. 2018. Deep Learning and Sentiment Analysis for Human-Robot Interaction. In *Europ. Semantic Web Conference*. Springer, 14–18.
[2] K. L. Cela, M. Á. Sicilia, and S. Sánchez. 2015. Social Network Analysis in E-Learning Environments. *Educational Psychology Review* 27, 1 (2015), 219–246.
[3] D. Dessì, G. Fenu, M. Marras, and D. Reforgiato Recupero. 2018. Bridging learning analytics and Cognitive Computing for Big Data classification in micro-learning video collections. *Computers in Human Behavior* (2018).
[4] D. Dessì, G. Fenu, M. Marras, and D. Reforgiato Recupero. 2018. COCO: Semantic-Enriched Collection of Online Courses at Scale with Experimental Use Cases. In *Trends and Advances in Infor. Systems and Technologies*. Springer, 1386–1396.
[5] G. Dragoni and M. Petrucci. 2017. A Neural Word Embeddings Approach for Multi-Domain Sentiment Analysis. *IEEE Trans. Affect. Comput.* 8, 4 (2017), 457–470.
[6] A. Dridi and D. Reforgiato. 2017. Leveraging semantics for sentiment polarity detection in social media. *Int. Jour. of Machine Learning and Cybernetics* (2017).
[7] G. Esparza, A. de Luna, A. O. Zezzatti, A. Hernandez, J. Ponce, M. Álvarez, E. Cossio, and J. de Jesus Nava. 2017. A sentiment analysis model to analyze students reviews of teacher performance using support vector machines. In *Int. Symp. on Distributed Computing and Artificial Intelligence*. Springer, 157–164.
[8] M. Giatsoglou, M. G Vozalis, K. Diamantaras, A. Vakali, G. Sarigiannidis, and K. Chatzisavvas. 2017. Sentiment analysis leveraging emotions and word embeddings. *Expert Systems with Applications* 69 (2017), 214–224.
[9] S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
[10] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov. 2016. Fasttext. zip: Compressing text classification models. *arXiv:1612.03651* (2016).
[11] Y. Li, Q. Pan, T. Yang, S. Wang, J. Tang, and E. Cambria. 2017. Learning Word Representations for Sentiment Analysis. *Cogn. Computation* 9, 6 (2017), 843–851.
[12] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. 2011. Learning Word Vectors for Sentiment Analysis. In *Proc. of the Annual Meeting of the Assoc. for Computational Linguistics: Human Language Technologies - Vol. 1*. 142–150.
[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).
[14] J. Pennington, R. Socher, and C. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.
[15] C. Raffel and D. P. Ellis. 2015. Feed-forward networks with attention can solve some long-term memory problems. *arXiv:1512.08756* (2015).
[16] D. Reforgiato, V. Presutti, S. Consoli, A. Gangemi, and A. Nuzzolese. 2015. Sentilo: Frame-Based Sentiment Analysis. *Cogn. Computation* 7, 2 (2015), 211–225.
[17] D. Reforgiato Recupero, E. Cambria, and E. Di Rosa. 2017. Semantic Sentiment Analysis Challenge ESWC2017. In *Semantic Web Challenges*. Springer, 109–123.
[18] E. Rudkowsky, M. Haselmayer, M. Wastian, M. Jenny, S. Emrich, and M. Sedlmair. 2018. More than Bags of Words: Sentiment Analysis with Word Embeddings. *Communication Methods and Measures* 12, 2-3 (2018), 140–157.
[19] B. Shi, Z. Fu, L. Bing, and W. Lam. 2018. Learning Domain-Sensitive and Sentiment-Aware Word Embeddings. *arXiv:1805.03801* (2018).
[20] D. Tang, F. Wei, B. Qin, N. Yang, T. Liu, and M. Zhou. 2016. Sentiment Embeddings with Applications to Sentiment Analysis. *IEEE Transactions on Knowledge and Data Engineering* 28, 2 (2016), 496–509.
[21] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin. 2014. Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*. 1555–1565.
[22] Z. Zhang and M. Lan. 2015. Learning sentiment-inherent word embedding for word-level and sentence-level sentiment analysis. In *2015 International Conference on Asian Language Processing (IALP)*. 94–97.

[4]https://code.google.com/archive/p/word2vec/
[5]https://nlp.stanford.edu/projects/glove/
[6]https://s3-us-west-1.amazonaws.com/fasttext-vectors/wiki.en.vec